An AI-driven Secure and Intelligent Robotic Delivery System

Wei Wang, Prosanta Gope, *IEEE Senior Member,* and Yongqiang Cheng

***Abstract*—Last-mile delivery has gained much popularity in recent years, it accounts for about half of the whole logistics cost. Unlike container transportation, companies must hire significant number of employees to deliver packages to the customers. Therefore, many companies are studying automated methods like robotic delivery to complete the delivery work to reduce the cost. It is undeniable that the security issue is a huge challenge in such a system. In this paper, we propose an AI-driven robotic delivery system, which consists of two modules. A multi-level cooperative user authentication module for delivering parcel using both PIN code and biometrics verification, i.e., voiceprint and face verification; Another non-cooperative user identification module using face verification which detects and verifies the identification of the customer. In this way, the robot can find the correct customer and complete the delivery task automatically. Finally, we implement the proposed system on a Turtlebot3 robot and analyse the performance of the proposed schema. Experimental results show that our proposed system has a high accuracy and can complete the delivery task securely.**

***Index Terms*—Robotic delivery system, AI, Cooperative user authentication, Non-cooperative user authentication, Multi-level authentication, Audio classification, Speaker verification, Face verification**

I. Introduction

Over the last decade, the emergence of new technologies made online shopping more accessible and widely acceptable. For example, the global retail sales traded online has doubled exceeding 10.7% in 2019 as compared to less than 5% in 2010 [1]. According to the report [2], 41% of US consumers receive one or two Amazon parcels weekly. The percentage is even higher among younger consumers – around 50% of respondents aged 18-25 and 57% of those aged 26-35 receive at least one parcel from Amazon per week. Delivery service is one of the most important tasks of running an e-commerce business, since customers expect a fast, reliable delivery service. As reported by [3], about 56% customers consider delivery option the most when shopping online. Over the course of e-commerce growth in the last decade, Amazon made two-day shipping the de facto industry standard [1]. China's JD.com, a Chinese e-commerce giant, says they make 90% of Chinese deliveries within 24 hours in 2018 [4]. Besides, online food delivery is another aspect which contributes to the increase of delivery services. The global online food delivery services market is expected to grow from $115.07 billion in 2020 to $126.91 billion in 2021 at a compound annual growth rate (CAGR) of 10.3%. Meanwhile, the market is expected to reach $192.16 billion in 2025 at a CAGR of 11% because of the companies resuming their operations and adapting to the new normal while recovering from the COVID-19 impact [5]. The last-mile delivery refers to the final step in the supply chain to delivery goods to the doors of customers, which is both the time-consuming and expensive part of the shipping process. The last mile of your product's delivery accounts for more than 53% of the total shipping costs [6]. It is reported that CAGR in the last-mile delivery sector over the next 10 years could exceed 14%, with the autonomous delivery segment projected to grow at over 24% from $11.9 billion in 2021 to more than $84 billion globally by 2031 [7].

Additionally, due to the outbreak of COVID-19, the demand for contactless treatment has expanded exponentially. The utilization rate of telemedicine service for all patients in the USA has increased to about 46% during the pandemic, which was only around 11% prior to the COVID-19 outbreak [8]. Furthermore, according to the Health Foundation report [9] of the UK, there is a severe shortage of health professionals, in particular staff shortages in clinics and lack of clinicians with specific expertise. Hence, using robots to deliver the prescribed medications autonomously to patients within hospitals will take these onerous tasks off hospital staffs, and free their valuable time for other more demanding tasks. Also, minimizing human contact with individuals who may have COVID-19 disease will reduce the risk of in-hospital disease spreading and enable high-risk health care workers to safely interact with patients through teleheath system.

A. Related work

For both delivery and healthcare applications, robots could be a promising way to solve the problems as mentioned above. Considerable number of companies have staked out urban landscapes to test autonomous last-mile delivery systems on the streets and mobile food delivery robots on campus sidewalks. For example, amazon Scout [10] is a fully-electric delivery system designed to safely get packages to customers using autonomous delivery devices. The Starship Technologies offers autonomous robots for stores, restaurants, and campuses [11]. JD's autonomous delivery robot successfully made its first delivery in Wuhan on Feb 6th, 2020 [12]. Moreover, nowadays many robots are used in medical centers (clinics) or hospitals. Robotic delivery of medicines in hospital wards using artificial intelligence techniques is investigated in [13]. Joy *et al.* [14] proposed a robot named MedRobo with some functionality of providing medicine as well as to measure the vital signs of the patient. In [15], a Smart Nursing Robot

W. Wang and P. Gope are with the Department of Computer Science, University of Sheffield, United Kingdom.

Y. Cheng is with the Department of Computer Science, University of Hull,United Kingdom.

was proposed to continuously monitor the patient and their medicine consumptions with timestamps. If necessary, the robot can be extended to support the delivery of medicines, food or clothes. Not only are large companies investing in research on autonomous delivery robots, but many research institutions are also researching related fields. For instance, Iurii [16] investigates a two-tier delivery network with robots operating on the second tier. It assumes that the first tier uses a conventional delivery truck travels to every robot hub to fill it with packages. Ostermeier [17] presents an approach for cost-optimal routing of a truck-and-robot system for last-mile deliveries with time windows, showing how to minimize the total costs of a delivery tour for a given number of available robots. Even though many robotic delivery systems have been proposed, authentication and security technologies specifically designed for the delivery are not fully developed. Protocols, such as [18], [19] which use special PUFs for security, are proposed in IoT areas. Yang [20] proposes a secure shipping infrastructure using robot which contains a cooperative user authentication module for delivering parcel using crypto primitives and a non-cooperative user identification module using Siamese Network for person re-identification. In their scheme, QR code is used for user authentication. However, owing to data leakage or spoofing, it is not secure to use only the QR code. In addition, the appearance of users changes over time, such as their clothing or hairstyle, which may affect the results of person re-identification.

B. *Our contributions*

In this paper, we propose an AI-driven secure and intelligent robotic delivery system which contains two modules: multi-level cooperative user authentication module and non-cooperative user authentication module. For the first module, we proposed a two-level cooperative user authentication approach. To achieve the multi-level authentication, a one-time password, i.e. PIN code, is used to check whether a customer can collect the parcel. Next, we do voiceprint or face verification to determine if this is the correct customer. In this scheme, the one-time password is used to gain the access to the next level data. In this method we can protect the biometric information of customers. Unlike multi-factor authentication approach that all factors must be input at the same time, only if the one-time password is correct, the robot runs the next level task and access the related data such as voiceprint or facial features. For the non-cooperative user authentication, face verification is used to find the correct customer. The robot can detect and extract a customer's face from video stream automatically, and then verify customer identity using face verification algorithm. If the robot find the correct customer, it will switch to cooperative mode to complete the delivery task, otherwise cancel the task for security.

The rest of this article can be organized as follows. In Section II, we provide a brief introduction to the techniques we used in this paper, such as one-time password, multi-level authentication, mel spectrogram, CNN model and XGBoost, etc. In Section III, we present our system model. The cooperative and non-cooperative user authentication module details will be introduced in section IV and V respectively. Experimental results are provided in Section VI. Finally, we conclude our article with concluding remarks in Section VII.

II. Preliminaries

A. *One-time Password*

A one-time password (OTP) is a password that is valid for only one login session or transaction [21]. Most enterprise networks, e-commerce sites and online communities require only a user name and static password for login and access to personal and sensitive data in the past years. Even though it is convenient, it is not secure due to the cyber security attacks like malware, phishing, keyboard logging, session hijacking, man-in-the-middle attacks, and other practices. To address the limitations of static passwords, an additional security credential is added to protect network access and customers digital identities. A temporary OTP is a reasonable choice. Normally, the OTP can be created using a secure hash function [22], such as a one-way hash function, based on the knowledge like order number, user ID or timestamp. It can be distributed using many ways, such as text messages, mobile phones or proprietary tokens.

B. *Multi-level Authentication*

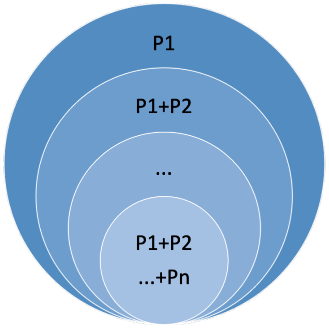
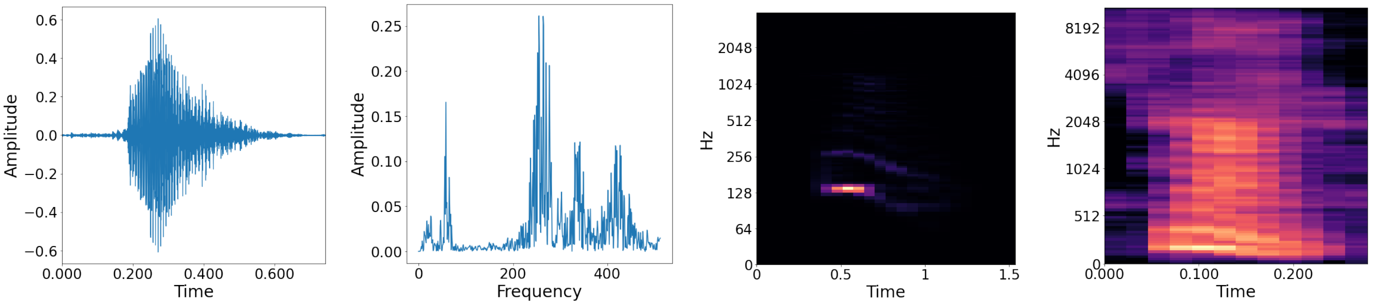


Fig. 1: Multiple level authentication.

Authentication is the act of proving an assertion. Many techniques have been proposed for this purpose, such as textual password, OTP or biometric authentication etc. Multi-factor authentication (MFA) is an electronic authentication method which uses two or more pieces of factors among the proposed authentications. A user is granted access to a website or application only after successfully presenting the required factors. An authentication factor represents some piece of data or attribute that can be used to authenticate a user requesting access to a system. Normally, the commonly used factors can be classified into three categories:

- Knowledge factor: Something only the user knows, such as a password or PIN code;
- Possession factor: Something only the user has, such as cards or smartphones;
- Inherence factor: Something only the user is, i.e., biometric data or behavior pattern.

Multi-level authentication (MLA) is a technique to authenticate data at multiple levels. Unlike MFA that all factors must



(a) Raw audio signal

(b) Spectrum

(c) Spectrogram

(d) Mel Spectrogram

Fig. 2: Audio Mel Spectrogram

be entered at the same time, in MLA, the privileges for a level are granted only if a user provides the correct password. According to the level of confidentiality of the data, users have different access rights.

Figure 1 illustrates the structural implementation of MLA. P1 is required to access the first level. In addition to $P1$, $P2$ must be entered to gain the privileges for the second level. Similarly, the whole system can be accessed when $P1$, $P2$ up to Pn are provided.

C. Mel Spectrogram

Mel spectrograms are spectrograms where the audio frequencies are converted to the mel scale. Sound is a variation in pressure which can travel through any medium (air, water, etc.) [23]. Audio data is obtained by sampling the sound wave at regular time intervals and measuring the intensity or amplitude of the wave at each sample. Figure 2a displays the signal in time domain, i.e. amplitude *vs* time. It gives us a sense of loudness or quietness of a clip at any point of time, but it gives us very little information about which frequencies the clip is presenting. Since a signal produces different sounds as it varies over time, its constituent frequencies, a.k.a. spectrum, also vary with time. Fourier transform is a mathematical transform that decomposes functions depending on time into functions depending on temporal frequency. Figure 2b shows the frequencies of a short time window. Normally, spectrogram is used to show the spectrum over overlapped short time windows as illustrated in Figure 2c.

Studies have shown that humans do not perceive frequencies on a linear scale. The mel scale provides a linear scale for the human auditory system, and is related to Hertz by the following formula, where m represents Mels and f represents Hertz:

$$m = 2595\ log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

As a result, we can get the mel spectrogram (Figure 2d) by converting the frequencies to the mel scale.

D. Convolutional Neural Network

Convolutional Neural Network (CNN) has been successfully used in a wide range of data analysis tasks, especially image processing in computer vision. It can automatically and adaptively learn spatial hierarchies of features, from low-level to high-level patterns. Figure 3 shows the overall architecture of CNNs consisting of four main types of layers: convolutional layer (Conv), activation function layer (Act), pooling layer (Pool), and fully connected layer (FC). Each layer in this deep architecture performs transformations by leveraging a number of convolutional kernels called filters. A standard CNN model has two key modules, the first module is used to extract features, the other one is in charge of classification.

Feature extraction: This module is a stack of blocks which is composed of convolution, activation and pooling layers.

Classification: This module is a traditional fully connected layer which takes in the learned representations from the feature extraction module and outputs a prediction.

E. XGBoost

XGBoost is originally proposed by Chen [24] in 2016. It is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. Even though artificial neural networks tend to outperform all other algorithms or frameworks in prediction problems involving unstructured data (images, text, etc.), decision tree based algorithms are considered the best-in-class when we are processing small-to-medium structured data. The reasons that the XGBoost algorithm performs so well is because it improves upon the base Gradient Boosting Machines framework through system optimization and algorithmic enhancements. In terms of system optimization, XGBoost approaches the process of sequential tree building using parallelized implementation. For example, it uses max_depth parameter as specified instead of criterion first to pruning trees which improves computational performance significantly. From the perspective of algorithmic enhancements, it penalizes more complex models through both LASSO and Ridge regularization to prevent overfitting. In addition to sparsity awareness, it employs the distributed weighted quantile sketch algorithm to effectively find the optimal split points among weighted datasets as well. The XGBoost library is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements a machine learning framework to provide a parallel tree boosting to solve many data science problems in a fast and accurate way [25]. In machine learning, the

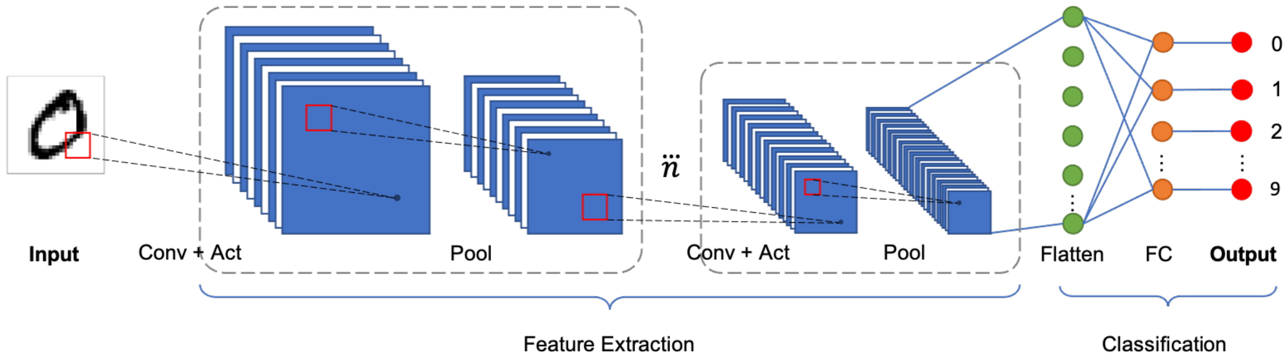


Fig. 3: Overall architecture of CNNs.

core definition of boosting is a method that converts weak learners to strong learners [26], and is typically applied to trees. More explicitly, a boosting algorithm adds iterations of the model sequentially, adjusting the weights of the weak-learners along the way. This reduces bias from the model and typically improves accuracy.

F. GhostNet

GhostNet was proposed by Han [27] in 2019. It is a specially designed CNN model that can be deployed on embedded devices which have limited memory and computation resources. A novel Ghost module was proposed in that paper to generate more feature maps from cheap operations. Based on a set of intrinsic feature maps, a series of linear transformations with cheap cost was applied to generate many ghost feature maps that could fully reveal information underlying intrinsic features. Compared with ordinary convolutional neural network, the total number of parameters and computational complexity required in the Ghost module are reduced without changing the size of the output feature graph. Taking advantage of the Ghost module, Ghost bottlenecks (G-bnecks) are designed to establish lightweight GhostNet. As shown in Figure 4, G-bneck mainly consists of two Ghost modules. The first one acts as an extension layer, increasing the number of channels. The second module reduces the number of channels to match the shortcut path. Then, shortcut is used to connect the input and output of the two Ghost modules. While the stride is 2, the shortcut path is implemented by a downsampling layer and a depthwise convolution (DWconv) with stride=2 being inserted between the two Ghost modules. Batch normalization (BN) and Rectified Linear Unit (ReLU) activation is applied after first Ghost module and DWconv, BN is used after the second Ghost module. Besides, a squeeze-and-excite module (SE) , which is a channel attention module proposed in [28], is added to certain G-bneck modules. Table I shows the overall architecture of GhostNet.

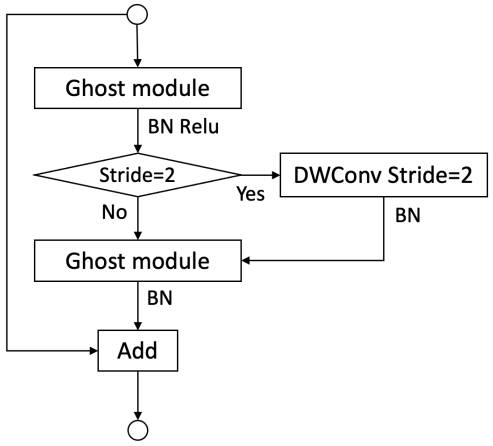


Fig. 4: Structure of Ghost bottleneck.

TABLE I: Overall architecture of GhostNet [27]. G-bneck denotes Ghost bottleneck. #exp means expansion size. #out means the number of output channels. SE denotes whether using SE module.

Input	Operator	#exp	#out	SE	Stride
$224^2 \times 3$	Conv2d 3×3	-	16	-	2
$112^2 \times 16$	G-bneck	16	16	-	1
$112^2 \times 16$	G-bneck	48	24	-	2
$56^2 \times 24$	G-bneck	72	24	-	1
$56^2 \times 24$	G-bneck	72	40	1	2
$28^2 \times 40$	G-bneck	120	40	1	1
$28^2 \times 40$	G-bneck	240	80	-	2
$14^2 \times 80$	G-bneck	200	80	-	1
$14^2 \times 80$	G-bneck	184	80	-	1
$14^2 \times 80$	G-bneck	184	80	-	1
$14^2 \times 80$	G-bneck	480	112	1	1
$14^2 \times 112$	G-bneck	672	112	1	1
$14^2 \times 112$	G-bneck	672	160	1	2
$7^2 \times 160$	G-bneck	960	160	-	1
$7^2 \times 160$	G-bneck	960	160	1	1
$7^2 \times 160$	G-bneck	960	160	-	1
$7^2 \times 160$	G-bneck	960	160	1	1
$7^2 \times 160$	Conv2d 1×1	-	960	-	1
$7^2 \times 960$	AvgPool 7×7	-	-	-	-
$1^2 \times 960$	Conv2d 1×1	-	1280	-	1
$1^2 \times 1280$	FC	-	1000	-	-

G. MobileFaceNet

MobileFaceNet [29] is an efficient lightweight CNN model. It uses less than 1 million parameters and is specifically tailored for high-accuracy real-time face verification on mobile and embedded devices. Its architecture is partly inspired by the

MobileNetV2 [30] architecture in which the residual bottlenecks proposed in MobileNetV2 are used as the main building blocks. PReLU is used as the activation function, since it has a better performance than using ReLU. To reduce the size of input feature maps, a fast downsampling strategy is used at the begining of the network, a linear 1×1 convolution layer following a linear global depthwise convolution layer as the feature output layer. Table II shows the detailed architecture of MobileFaceNet.

TABLE II: Overall architecture of MobileFaceNet [29]. *T* means expansion size. *C* means the number of output channels. Each layer repeates *N* times. The first layer of each sequence has a stride *S* and all others use stride 1.

Input	Operator	T	C	N	S
112^2 × 3	Conv2d	-	64	1	2
56^2 × 64	DW Conv2d	-	64	1	1
56^2 × 64	bottleneck	2	64	5	2
28^2 × 64	bottleneck	4	128	1	2
14^2 × 128	bottleneck	2	128	6	1
14^2 × 128	bottleneck	4	128	1	2
7^2 × 128	bottleneck	2	128	2	1
7^2 × 128	Conv2d 1×1	-	512	1	1
7^2 × 512	AvgPool 7×7	-	512	1	1
1^2 × 1280	Linear Conv2d 1×1	-	k	1	1

H. Spatial Group-wise Enhance

Spatial Group-wise Enhance (SGE) is a lightweight attention module [31] which is presented by Li *et al.* [32] in 2018. Compared to other attention modules, SGE has fewer parameters, less computational complexity, and a more interpretable mechanism.

One of its important highlights is that it can also achieve strong gains in classification and detection performance without increasing the amount of parameters and calculations. At the same time, compared with other attention modules, it is the first generation source that uses the similarity between local and global as the attention mask, and has a very strong semantic representation and enhanced interpretability.

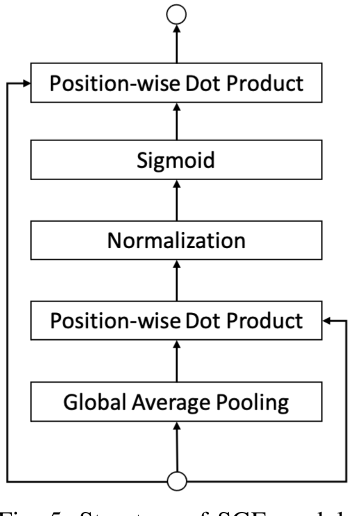


Fig. 5: Structure of SGE module

I. Cosine Similarity

Cosine similarity measures the similarity between two non-zero vectors of an inner product space [33]. It is defined to equal the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is generally used as a metric for measuring distance when the magnitudes of the vectors are negligible. For example, it is widely used to measure document similarity in text analysis.

As shown in Figure 6, let x and y be two vectors for comparison. Using the cosine measure as a similarity function, we have

$$S_{cos}(x,y) = \cos(\theta) = \frac{x{\cdot}y}{||x||||y||} \tag{2}$$

where $||x||$ is the Euclidean norm of vector $x = (x_1, x_2, \cdots, x_p)$, defined as $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$. Conceptually, it is the length of the vector. Similarly, $||y||$ is the Euclidean norm of vector y. The measure computes the cosine of the angle between vectors x and y. The cosine similarity is bounded in the interval $[-1, 1]$ for any angle θ. For example, two vectors with the same orientation have a cosine similarity of 1, two vectors oriented at right angle relative to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1.

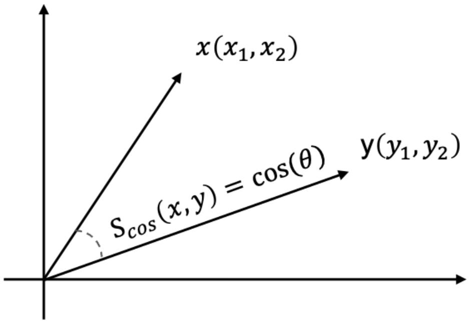


Fig. 6: Cosine Similarity.

III. System and adversary model

In this section, we first describe our system and and adversary model. Subsequently, we describe our proposed scheme.

A. System model

Figure 7 shows the system model of our proposed scheme, in which we consider three main entities: a client, a server and a robot. In our system model, the client is a customer who sends requests to the server and receives messages from it. For instance, a customer can use a digital device, such as mobile phones, tablets or personal computers etc., to place an order online and receive the required messages, i.e. QR code, PIN code or delivery time etc., in order to prove his/her legitimacy and then collect the parcel from the delivery robot.

On the other hand, the server accepts orders from clients over the Internet and gives carriage directions in offline warehouses. It requires to fulfil the delivery requirements, including confirming orders, distributing messages for authentication to the client as well as transmitting key information, such as extracted customer pre-registered voiceprint feature, to the robot using the secure channel. It is also the core part of the communication process. The communication between the server and the client will take place via the public Internet whilst private network of the server is used for message exchanges between the server and the robot.

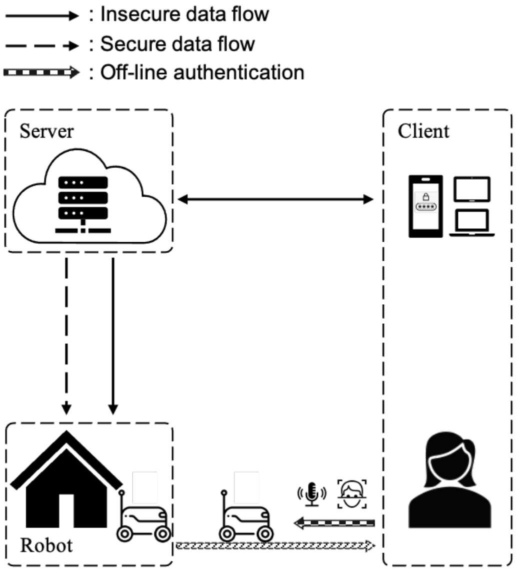


Fig. 7: System model for the proposed system.

The working principle of the proposed system can be divided into the following steps:

Step 1: A customer places an online-order.

Step 2: The server receives the request and then assign a robot to delivery the parcels. After that, the server prepares the related information (which will used for verifying each others) both client and robot.

Step 3: The robot loads data from the server via security networks and navigates to the destination automatically.

Step 4: The customer needs to provide the valid PIN code in order to collect parcel from the robot. Meanwhile the robot also use the customer's voice and face recognition to identify the customer. If the PIN code is correct and one of the biometirces is verified, the delivery complete. If the PIN code does not match, the robot will switch to non-cooperative mode of verification i.e., facial verification to detect and check the customer's identification.

B. *Off-line authentication approaches*

Therefore, to complete the delivery process the robot runs both cooperative off-line authentication and non-cooperative off-line authentication methods. Figure 8 illustrates the details of the proposed scheme. For the first part, the customer triggers the process when the robot arrives. It records the customer's voice and recognizes the PIN code. The audio will be split into segments, and then classified to digits which is range from 0 to 9. If the PIN code is correct, the robot will verify the customer identification by checking the cosine similarity of voiceprint features or facial image features. If the verification is successful, then we can say that the delivery assignment has been completed successfully. Otherwise, the person will be treated as an illegitimate customer, the robot will cancel the delivery job. If the PIN code does not match, it will repeat the record voice and recognize PIN code steps as mentioned above until it reaches the max iterations. As the deliver assignment is time limited, the robot will switch to non-cooperative off-line authentication mode when the system does not reach the deadline. Under this mode, the system will execute the face detection and recognition. It will continue to detect and recognize customer's face from video stream until session has expired or the customer has been found. The robot will cancel delivery assignment if the session has timeout, otherwise, it will re-run the cooperative off-line authentication.

C. *Adversary model*

In our adversary model, we consider the server as a trusted entity. Usually, they are owned by private companies or organizations themselves and are deployed locally or on the cloud end, such as Amazon, a hospital or a medical center. Both the client and the server communicate via insecure public network, and part of communications between the Robot and Server is insecure as well. Hence, there is a possibility for an adversary to eavesdrop information, leading to the data leakage. Additionally, the proposed scheme uses biometrics to verify customer's identification, where an adversary may pretend to be a legitimate client and cheat the robot. Therefore, it is important to consider authentication properly.

IV. Multi-level cooperative authentication

In this section, we present our cooperative off-line authentication scheme which uses MLA. Figure 9 shows the outline structure of the proposed scheme. It contains two stages, including PIN code identification and identity identification (i.e. voiceprint and face identification), which forms a two-levels (two-factors) off-line authentication system for cooperative users. When the robot arrives at the destination, it will wait until the authentication process is triggered by certain method. For example, a user can manually click the pickup button on the screen of the robot. We will first introduce the implementation of the two stages in details, and then explain how to complete the authentication.

A. *PIN Code Identification*

In this part, we will introduce how the system completes the PIN code identification. We introduced CNN and XGBoost in section II. The structure of a typical CNN network includes two parts: feature extraction and classification which is shown in Figure 3. XGBoost is a classifier with better performance and can often get higher accuracy than random forest or logistic regression etc. PIN code is a widely used authentication method in the industry which usually appears in the form of 4 or 6 digits. Here, we use a combination of 4 digits as the PIN code. Hence, we can regard the recognition of each digit as a classification task which has 10 classes.

In audio classification, a common way is to convert the original audio into a spectrogram, and then classify the image.

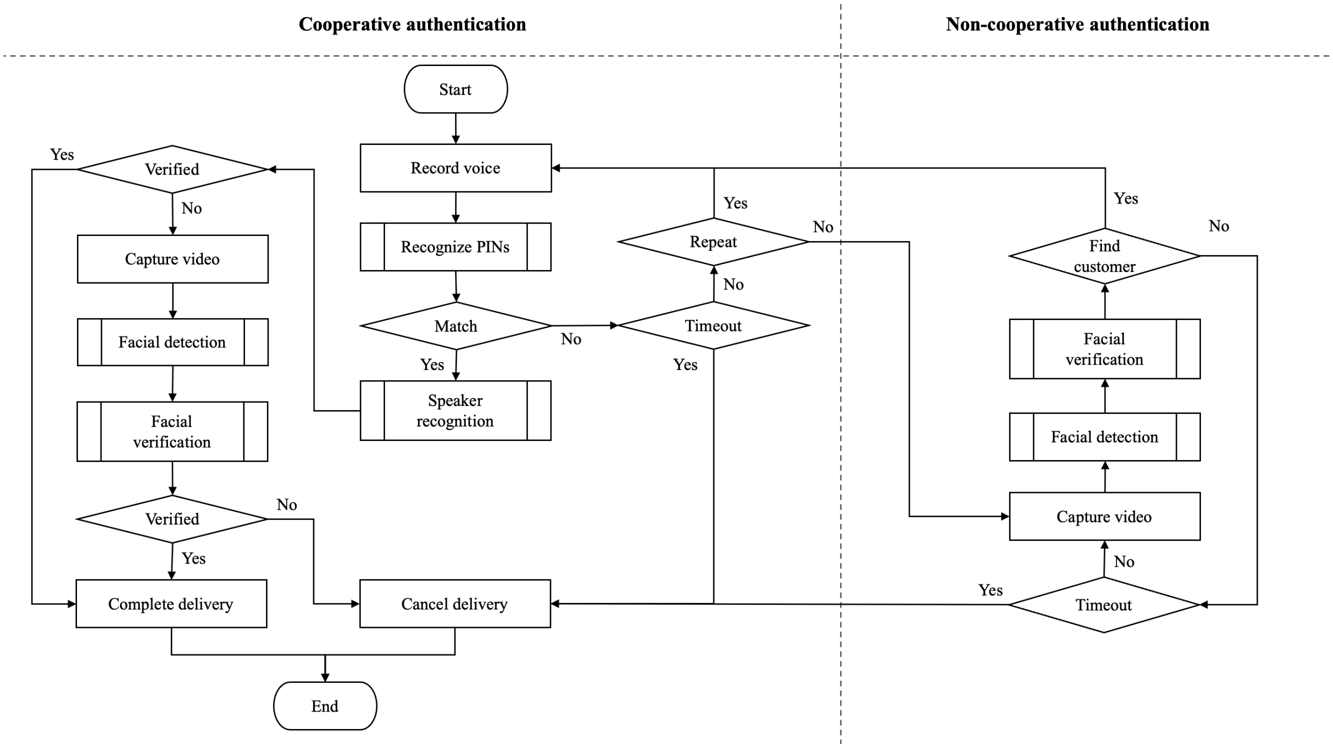


Fig. 8: The authentication approaches of the proposed system.

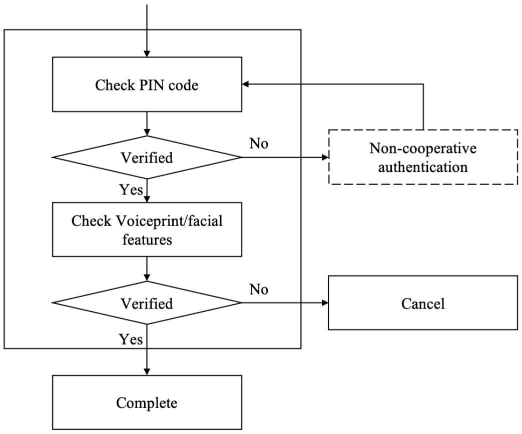

Check PIN code
Verified
No
Yes
Non-cooperative authentication
Check Voiceprint/facial features
Verified
No
Yes
Cancel
Complete

Fig. 9: The outline of the cooperative authentication.

Normally, a CNN network is used for feature extraction. Generally speaking, we can add a fully connected layer to the CNN network for classification, but the accuracy may not be high, and the fully connected layer contributes to most of the parameters. In order to improve the classification accuracy and the calculation speed, we use XGBoost to classify the extracted features.

1) Feature extraction: Considering the model calculation, the structure of CNN for image classification consists of 6 convolution layers, a flatten layer and 3 fully connected layers. Each convolution layer follows an activation layer and a pooling layer. The CNN network will be trained independently, and then extract the output of the second full connection layer as the feature vector. Configuration of feature extraction architecture is shown in Table III.

TABLE III: Configuration of CNN feature extraction architecture. *K* is kernel size. The stride of each layer is *S*. *P* denotes the padding size. *C* means the number of output channels.

Layer	Type	K	S	P	C
Data	Input	-	-	-	3
Conv 1	Convolution	3×3	1	1	16
Pool 1	Max pooling	2×2	1	1	16
Conv 2	Convolution	3×3	1	1	32
Pool 2	Max pooling	2×2	1	1	32
Conv 3	Convolution	3×3	1	1	64
Pool 3	Max pooling	2×2	1	1	64
Conv 4	Convolution	3×3	1	1	128
Pool 4	Max pooling	2×2	1	1	128
Conv 5	Convolution	3×3	1	1	128
Pool 5	Max pooling	2×2	1	1	128
Conv 6	Convolution	3×3	1	1	256
Pool 6	Adaptive average pooling	1×1	1	1	256
Mapping	Fully connected	-	-	-	128

Steps of feature extractor as follow:

1. Initialize the parameters of the CNN network.

2. After convolution calculating, feature data obtained through each activation layer and pooling layer.

3. The feature map forms a one-dimensional vector being processed by fully connected layers.

4. Vectors initialized into a new training data set which is used for predicting by subsequent classifier.

2) Classification: We use the dataset generated by the CNN network to train the XGBoost model, to obtain a tree structure suitable for feature classification.

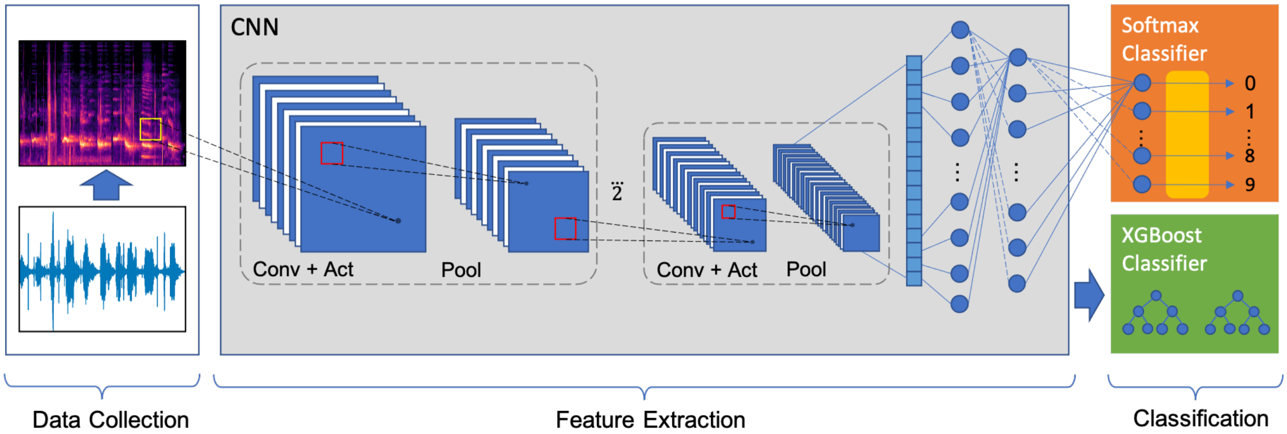


Fig. 10: The structure of CNN-XGBoost hybrid model.

XGboost provides a large number of hyperparameters for tuning. In this article, we focus on several parameters that have a greater impact on the results: *max_depth*, *min_child_weight*, *subsample*, *colsample_bytree*, *eta*. As it is a multi-classification task, *objective* is *"multi:softmax"*, *num_class* is 10, and *eval_metric* is *["merror", "mlogloss"]*.

3) Training the CNN-XGBoost hybrid model: In the proposed training scheme, we first train the CNN model, the configuration is defined in Table III, and then train the XGBoost classifier using the dataset generated from the trained CNN model. The CNN model uses Mel spectrograms as input data and outputs the predicted classification results. The features are extracted through the first 6 convolutional layers. A 256×1 dimensional feature vector is converted through the flatten layer, and then input the following fully connected layer for classification, the activation function of the last layer is Softmax. We uses CrossEntropyLoss as the loss function, the optimizer is Adam, and learning scheduler is MultiStepLR. To train the CNN model, the dataset is split into training and testing dataset at a ration of 80:20. We use the the training dataset to train the model for multiply epochs, and data in each epoch is split into batches by predefined $batch\_size$. When the training of the CNN model is completed, we extract the output of the second fully connected layer as the training data to train the XGBoost model. For the XGBoost model, optimizing hyperparameters is the main goal of model training. As XGBoost provides a large number of hyperparameters for tuning, we only fine tune the most important parameters which are listed in the previous section. Ray Tune [34] is a flexible, high-performance distributed execution framework which is used for hyperparameter optimization, we use it to tune the XGBoost model. Grid search is used to fine tune the hyperparameters, the search space is shown in Table IV. The scheduler is $ASHAScheduler$ which will enable aggressive early stopping of bad trials and the max training iteration is 100. To train the XGBoost model, the dataset is divided into training and testing datasets at an 80:20 ratio as well. Then we use the CNN model to generate the features which form the new training and testing datasets for the XGBoost.

TABLE IV: XGBoost grid search space using Ray Tune and the chosen value.

Hyperparameter	Search space	Value
objective	multi:softmax	multi:softmax
num_class	10	10
eval_metric	["merror", "mlogloss"]	["merror", "mlogloss"]
max_depth	tune.randint(3, 11)	6
min_child_weight	tune.choice([0.1, 0.3, 1, 3, 10, 30, 100])	0.1
subsample	tune.uniform(0.5, 1.0)	0.58783
colsample_bytree	tune.uniform(0.1, 1.0)	0.95178
eta	tune.loguniform(1e-4, 1e-1)	0.00031

B. Voiceprint Identification

This section mainly introduces the realization of user identification in the proposed cooperative offline authentication scheme. In this paper, we use both voiceprint verification and face verification, and they are runned in serial order. As face verification is also applied to the non-cooperative offline authentication mode, its implementation will be introduced in section V, the details of voiceprint verification will be introduced in this part. The voiceprint verification system is an application system based on the speech identification of the speaker. It is a technical system that automatically recognizes or authenticates the speaker's identity according to the physiological and behavioral characteristics of the speaker characterized by the voice information. Based on the different applications of the voiceprint recognition system, the technical implementation of the voiceprint verification system can basically be divided into two categories: speaker confirmation and speaker recognition technology. The former is used to determine whether an unknown speaker is the designated person (1:1 Recognition System); the latter is used to identify which of the recorded speakers is the unknown speaker (1:N Recognition System). In this paper, as we are delivering parcels to a certain customer, the customer's voiceprint features can be collected in advance, such as during user registration, we can consider it to be a speaker confirmation technology.

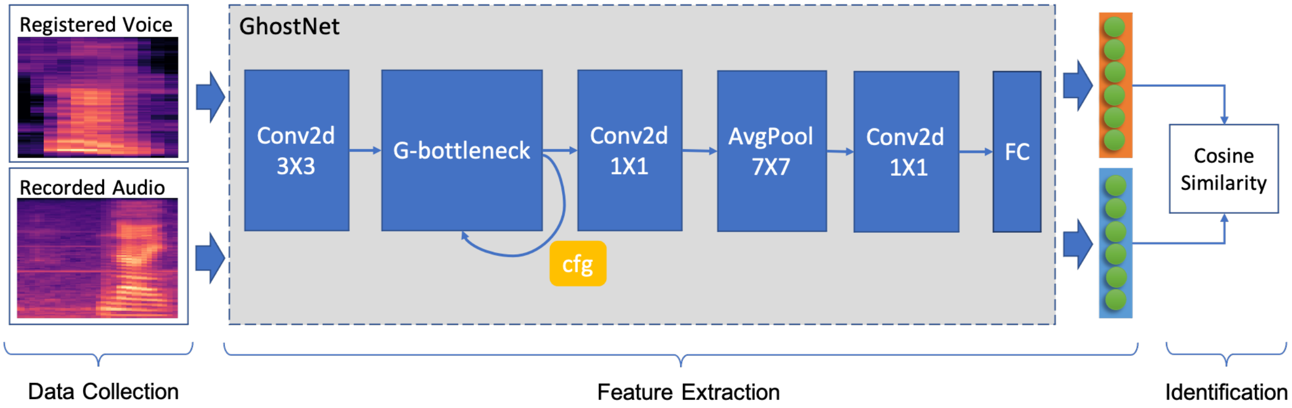


Fig. 11: The structure of Ghostnet voice identification model.

From the perspective of the use of the voiceprint verification system, the source of the sound that needs to be identified can be further divided into three categories, namely text-prompted, text-related and text-independent. Among them, the text-prompted voiceprint recognition system requires that the person being authenticated needs to provide the pronunciation of the given text; the text-related voiceprint recognition system requires the system to record a certain number of a customer's voices that specify the content of the text; while the text-independent voiceprint recognition system does not specify the content of the speaker's pronunciation, as long as the speaker's voice is recorded in the system, it can be identified whether it belongs to the speaker. The used 4 digits PIN code is randomly generated, so it is a text-independent voiceprint recognition system.

Voiceprint recognition mainly includes voiceprint feature extraction and identification.

1) Feature extraction: GhostNet is introduced in section II. It is a lightweight CNN network model specially designed by Huawei for devices with limited resources on the mobile terminal. It has better performance than the MobileNetV2 model. We use this model for voiceprint feature extraction.

Steps of feature extraction as follow:

1. Initialize the parameters of network.

2. Take the preprocessed data as input, after convolution calculating, feature data obtained through each G-bneck layers. Channels increased to 1280 after 2 following convolutional layers.

3. The feature map forms a one-dimensional vector being processed by a fully connected layer.

2) Identification: For the purpose of identification, we compare the cosine similarity with a predefined threshold. First of all, we collect the customer's audio record. Since the robot has recorded the audio, the system can reuse the audio in this phase. Second, the robot extracts the user's voiceprint features using the trained Ghostnet model. Third, it calculates the cosine similarity with the voiceprint features registered by the customer using Euqation 2. Finally, it checks whether the similarity exceeds a certain threshold to verify the user's identity.

3) Training the GhostNet model: We use the same configuration of GhostNet model shown in Table I, but change the output feature size to 512. Here, the Mel spectrogram is the input data as well. However, we extract the voiceprint feature in this part, so we can use a longer audio compared with the classification model. For this model, the input size is 224×224×3. ArcFace [35] is used as the loss function. To train the model, we preprocess the training and test dataset first. For each record, we convert it to multiple utterances of fixed size 1.2 seconds with 50% overlap. We use the training dataset to train the model for multiply epochs, and data in each epoch is split into batches by predefined $batch\_size = M \times N$ where M represents number of speakers in batch and N stands for number of utterances per speaker.

C. Authentication

Since we have explained how to train the CNN-XGBoost hybrid model and GhostNet model which are used in the two stages, we will introduce the way we use them to complete the authentication. Except the pretrained models, the robot loads the customer related data, such as PIN code, customer's voiceprint feature vector and customer's facial feature vector etc., from the server when it departures from the warehouse.

The first stage checks whether the PIN code is correct. If the PIN code is correct, it starts the next stage. Otherwise, the user is allowed to retry 2 more times. If the customer cannot input the PIN code correctly within the limited number of times, it will switch to the non-cooperative authorization mode which will be described in detail in section V.

The second stage verifies the user identification using the voiceprint recognition or face recognition. Since the robot has collected the user's voice and face image in the first stage, we can re-use the audio file to extract voiceprint features or use the face image to extract facial features for user identification. Anyone or both of them, which is chosen based on the tradeoff of security and delivery rate, passed, we consider the user's identity to be confirmed, and the delivery task complete successfully. Otherwise, the robot will cancel the delivery. Because in this situation, we believe that the PIN code has been leaked.

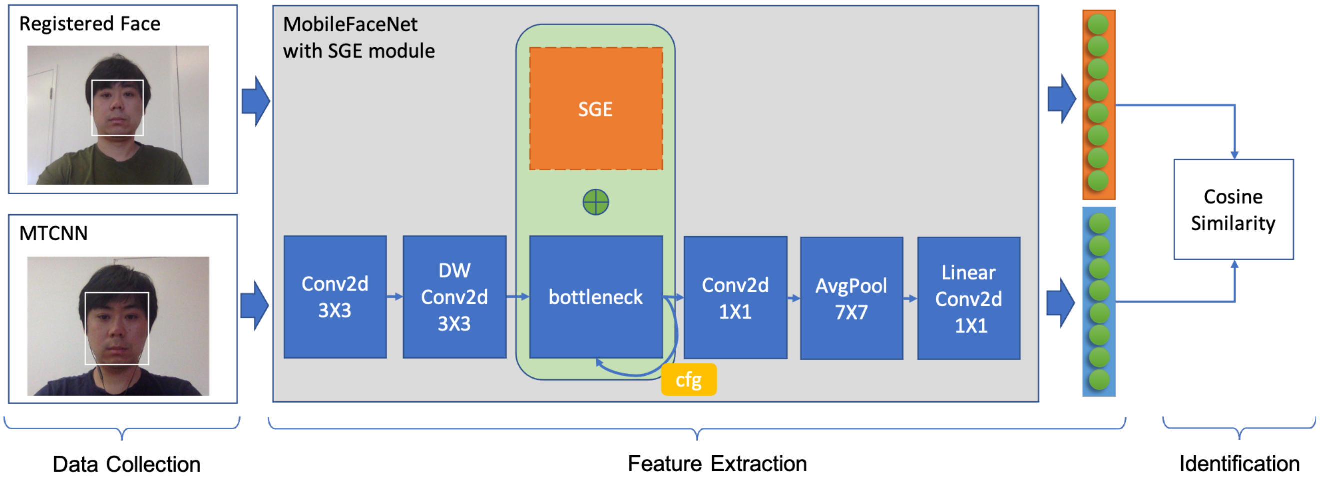


Fig. 12: The structure of improved MobileFaceNet model with SGE module.

D. *Correctness analysis of the proposed scheme*

As introduced in section III-C, we consider the server as trusted entity and part of communications between the robot and server is secure, so the data such as PIN code, voiceprint features and facial features that are loaded to the robot is safe. However, the client and the server communicate via insecure public network, an adversary may eavesdrop information which leads to the data leakage. For this part, even though the data is transmitted via insecure public network, we can still prevent the adversary to eavesdrop information by encrypt transmission data. The data can be decrypted in the application on the customer's mobile phone. Meanwhile, the customer can request the PIN code from the server only after the robot has set off. In this way, the adversary does not have enough time to intercept the delivery robot. We assume that adversaries got the PIN code, they can only pass the first level of verification. The system checks the voiceprint in the second level, it is hard for the adversary to collect the audio since the PIN code is randomly generated. If they use their own voice, they cannot be recognized by voiceprint recognition. Moreover, in this project, we combine voiceprint recognition and face recognition on the second level to improve security. Since we use multiply machine learning algorithms to check the correctness of the customer. The total accuracy of the result relies on the accuracy of the algorithms, more details will be discussed in section VI-C4.

V. Non-cooperative facial recognition

In this section, we introduce the non-cooperative recognition system to meet the challenge that the client is not active in presenting the PIN code. This proposed scheme is a supplement of the cooperative authentication. When the task of PIN code recognition fails for a maximum times, the robot will switch to this mode to detect and identify the nearby client. This part is subdivided into modelling phase and identification phase. The former modelling phase is done before the delivery request arriving at the server, including data pre-processing and model training. Then, the identification phase takes place upon receiving the order, which consists of preparation of comparison, face detection and extraction, resizing image and facial recogination. If succeeded, the robot will go toward the target client to remind him to tell the PIN code to it and then continue cooperative authentication process.

A. *Modeling phase*

To use the face identification, the company must collect the face image of clients. Normally, this can be done during registration and the robot can also save a customer's face image through the completed deliveries. These images can be used as a larger training dataset to train or test the improved MobileFaceNet (MobileFaceNet-SGE) model. Obviously, the training process is completed before orders begin. We will demonstrate the details in the following paragraphs.

First of all, we should prepare for the dataset. In this paper, we use the CASIA-Webface [36] as the training dataset which contains 494,414 images of 10,575 people, with a size of 112×96 pixels. The test dataset is LFW [37]. It contains 13,233 face images of 5749 people, with the same size.

Secondly, in model training stage, we apply the training dataset to the MobileFaceNet-SGE to train the model. The structure of the MobileFaceNet-SGE is displayed in Figure 12, the configuration of bottleneck modules (cfg) can be found in Table II. The process of training is further explained as follows:

In the proposed training scheme, we use the the training dataset to train the model for multiply epochs, and data in each epoch is split into batches by predefined $batch\_size$. When we start the training task, the DataLoader randomly select $batch\_size$ images from the remaining set in the current epoch. These images are fed to the model to extract the features. It accepts a 112×96 pixel image with a depth of 3. Through a convolutional layer and a depth-wise convolutional layer, the depth increases to 64. Then following 5 layers of bottleneck modules, the depth increases to 128. A SGE module, which has the same parameter configuration, is added to each bottleneck and the groups number of a SGE module is 64. The depth is further expanded to 512 using a convolutional layer with kernel size 1×1. In this way, the model can generate more features. The next layer is different from a flatten layer of general neural networks. It uses a 7×7 convolutional network for linear transformation, and finally reduces the feature dimension to 128. The weights parameters are updated based on the ArcFace loss function.

B. Designs of improved MobileFaceNet using SGE

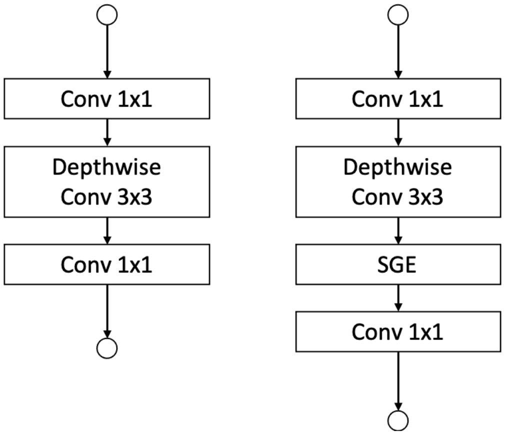


Fig. 13: MobileFaceNet-SGE bottleneck layer structure.

We introduced MobileFaceNet and SGE in Section II. This section will introduce how to use SGE module to improve MobileFaceNet network.

As shown in Figure 12, MobileFaceNet contains 5 bottleneck layers, and the configuration can be found in Table II. The structure of SGE module is shown in Figure 5. In Figure 13, an original bottleneck module is shown on the left, from the figure we can find out that the bottleneck contains 3 layers: one convolutional layer, one depth-wise convolutional layer and another convolutional layer. On the right, it shows the structure of a bottleneck with a SGE module. Here, we add the SGE module after the depth-wise operation of the bottleneck layer, to learn the correlation between the group channels, and filter out the attention to the group channels. Although the amount of calculated parameters is slightly increased, the performance of the model is improved as well.

C. Identification phase

In the identification phase, to determine whether the customer is the correct one, as show in Figure 12, the robot follows three steps: data collection, feature extraction and identification. We assume that the robot has loaded the pre-trained model and related data from the server when it departures from the warehouse. Once the customer is detected, the robot switches to the cooperative mode.

1) Data Collection: For the first step, the robot must capture customer's face in real time when it arrives at the destination. When the robot switches to non-cooperative mode, it will start the camera and use MTCNN algorithm [38] to detect faces from the video stream. The MTCNN will return the position of the detected face, the facial image can be extracted according to the returned position from the frame. And then the extracted facial image is scaled to 112×96 pixels.

2) Feature Extraction: In this step, the robot extracts the facial features by using the pre-trained MobileFaceNet-SGE model. From Figure 12 we can find out that both feature vectors are generated from the same model. In the data collection step, two face images are collected. The registered face is provided by the customer on the website in security environment, while the second input data is collected in the previous step using MTCNN algorithm in real time. For the MobileFaceNet-SGE model, each bottleneck module is combined with a SGE module to improve the performance. Two feature vectors are generated based on the input data. Generally, the registered face can be processed in advance on the server, which can avoid data leakage and speed up execution. The robot can load the features via security connection in the warehouse as shown in Figure 7.

3) Identification: Since we have extracted the feature vectors from the first two steps, the robot calculates the cosine similarity of the feature vectors to determine whether this is the correct customer in this part. The cosine similarity is calculated using Equation 2. If the captured face image belongs to the correct customer, the cosine similarity is close to 1, otherwise it is close to 0 or negtive. In this paper, we use a pre-defined threshold to judge whether this is the correct customer.

VI. Discussion

In this section, we first compare our proposed method with respect to our previous work [20] and then provide our experimental outcomes to evaluate the performance of the algorithms used in the proposed schemes that includes the voice classification module based on the CNN-XGBoost hybrid network, the voiceprint recognition module based on the GhostNet network and the face recognition module based on the MobileFaceNet-SGE network.

A. Comparison with our previous work

Table VI shows the comparison with previous work. From the table we can find that both of them should collect customer's privacy data, such as voice audio and face image. However, our proposed scheme can protect compromised client's device or account, and it is crypto primitives independent. Moreover, we use more factors for authentication which means that the proposed scheme is more secure than the previous one. The non-cooperative authentication has higher availability as a person's face does not change much in a few years while changes clothes everyday.

B. Experimental environment

All implementions in the experiment are developed using python based on the pytorch framework. The python version is 3.7, and pytorch version is 1.8.1. Training and testing were carried out on the High Performance Computing (HPC) platform in the University of Sheffield, running on NVIDIA K80 GPU. Table VIII shows the requested resources in this project. According to the rules of the platform, we need to apply for one GPU for each CPU core. The CPU cores

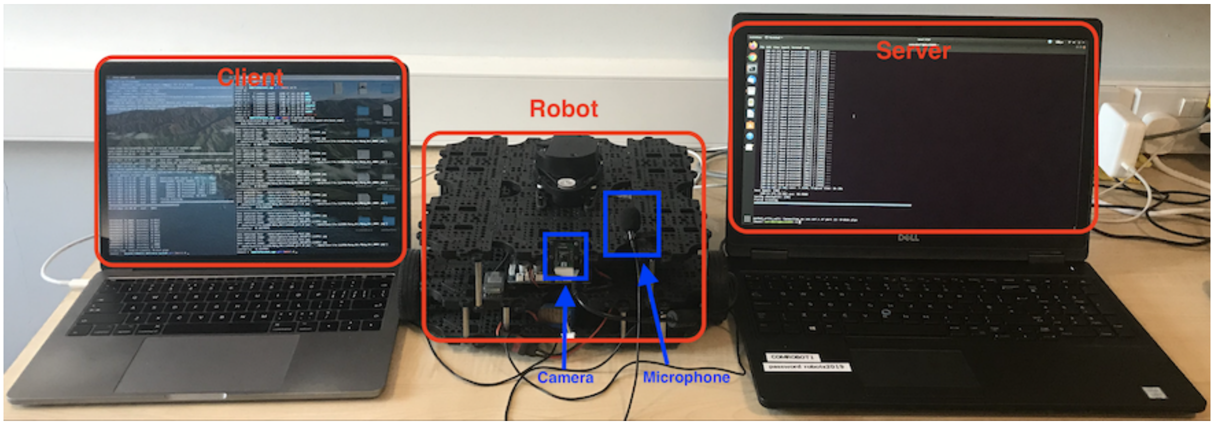


Fig. 14: The real-life experimental platform.

TABLE V: Implementation Environment.

Terminal	Machine	System	Python version	IP address
Client	Laptop	MacOS 11.04	3.7	192.168.31.218
Server	Laptop	Ubuntu 18.04	3.7	192.168.31.93
Robot	Robot (TurtleBot3)	Ubuntu 18.04	3.7	192.168.31.94

TABLE VI: Compare with previous work.

Scheme	Customer's privacy	Protection against compromised client's device or account	Crypto primitives	Cooperative multi-factor authentication	Non-cooperative authentication availability
Yang *et al.* [20]	Yes	No	Dependent	One factor	Medium
Proposed scheme	Yes	Yes	Independent	Two factors	High

TABLE VII: Configuration of Deep Learning network hyperparameters.

Model	Epoch	Init learning rate (LR)	Batch size	LR change epoch (×0.1)	Weight decay	Stochastic gradient descent strategy	momentum
CNN	50	0.001	64	25,37,43	4e-5	Adam	-
GhostNet	50	0.001	32	30	5e-4	SGD	0.9
MobileFaceNet+SGE	50	0.1	256	25, 37, 43	4e-5	SGD	0.9

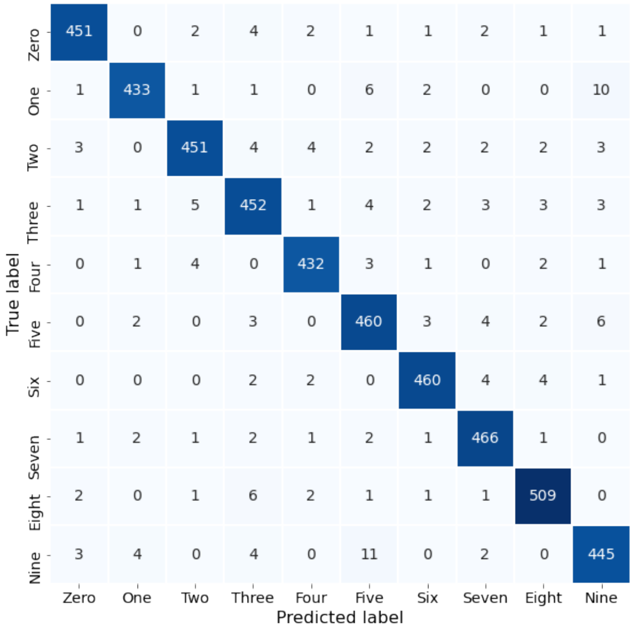


Fig. 15: Confusion matrix.

TABLE VIII: Requested hardware resources for each model.

Model	CPU cores	GPU per CPU core	GPU onboard memory
CNN	2	1	4G
XGBoost	3	1	8G
GhostNet	3	1	4G
MobileFaceNet+SGE	4	1	4G

and GPU onboard memory of each training task is slightly different, these parameters are based on the dataset size and training time.

As shown in Figure 14, Turtlebot3 is used as the delivery robot, and all the trained models are running on this platform. Operating system of the robot is Ubuntu 18.04. The voice is recorded using a USB microphone, and librosa 0.8.1 [41] is used to process the recorded audio, such as splitting audio into segments, converting audio to Mel spectrogram, etc. The image is captured from the camera using OpenCV. Besides, we use two laptops as the server and client respectively. The operating system of the client is MacOS 11.04 while the Server's is Ubuntu 18.04. The configuration of all platforms are shown in Table V. We use python virtual environment to

TABLE IX: Datasets.

Model	Training dataset	Test dataset	Split ratio (training/testing)
CNN+XGBoost	Speech Commands [39]	Speech Commands	80/20
GhostNet	TIMIT [40]	TIMIT	80/20
MobileFaceNet+SGE	CASIA-Webface [36]	LFW [37]	-

run all python scripts. Hyperparameter configuration for deep learning models and XGBoost model are shown in Table VII and IV respectively. Since we tune the XGBoost model using Ray Tune which is introduced in section IV-A, part of the hyperparameters like eta, $subsample$ and $colsample\_bytree$ are generated randomly, we can find this out from the values of them.

C. Result

Average accuracy is one of the most widely used evaluation index for classification evaluation. It represents the overall performance of the model and reflects the overall level of classification ability. In order to test the performance of the models proposed in this paper, we evaluate it on the databases listed in Table IX. All networks are trained on the original training datasets. As the PIN code is consists of digit numbers only, we only selected a subset of the whole commands, i.e., 0, 1 up to 9. There is only one dataset for the CNN-XGBoost hybrid model and GhoseNet respectively, 80% is used to train the model and the rest is used for testing. Meanwhile, the CASIA-Webface dataset is applied to train the MobileFaceNet and LFW is used for testing, so we do not need to split the dataset.

TABLE X: CNN-XGBoost accuracy.

Model	Accuracy	Accuracy with white noise
CNN	94.0%	93.74%
CNN+XGBoost	96.42%	96.22%

1) Test of PIN code classification: The training and testing data were preprocessed in the same way, and then used to train the CNN model. Next, we use the trained CNN model to extract the feature vector from the training dataset. It was fed to the XGBoost model. Finally, we compared the classification accuracy of the CNN model and the XGBoost model. From Table X we can find that the accuracy of the XGBoost model was improved by about 2.42% compared to the CNN model. Figure 15 shows the confusion matrix of XGBoost classification results. It can be seen from the results that the number of misclassifications of 1 as 9, and 9 misclassifications as 5 is high. This is mainly because the waveplot of them is very similar, so the converted mel spectrogram image is similar. Due to the limitation of the training dataset size, they cannot be effectively distinguished. Besides, since the dataset contains background noise, we add white noise to the original audio to generate new training and testing dataset. The result shows that white noise has a slight impact on the algorithm, mainly because the CNN algorithm has the ability to reduce the influence of noise.

2) Test of voiceprint verification: We preprocess the dataset first, and then feed the preprocessed image to network after resizing the image to 224×224. As Resnet is a commonly used backbone and GhostNet is a lightweight model, we choose Resnet34 which has smaller parameters than other Resnets to compare the performance with our proposed algorithm. Besides, to compare with the state-of-the-art, a RW-CNN model proposed in [42], DNN model proposed in [43] are also considered. Table XI shows the accuracy results of them, from the table we can see that GhostNet has the best accuracy result. The accuracy is about 1.49% higher than the Resnet34, and 0.33% higher than the DNN model. Besides, white noise is considered as well. The result of this experiment showed the same result as the above experiment. The white noise does not impact the algorithm a lot. The GhostNet model is about 0.62% higher than the RW-CNN model. However, we also need to consider that the CNN model trained in [42] only use a subset of TIMIT dataset.

TABLE XI: GhostNet accuracy.

Model	Accuracy	Accuracy with white noise
GhostNet	96.98%	96.62%
Resnet34	95.49%	95.04%
RW-CNN [42]	-	96.00%
DNN [43]	96.65%	-

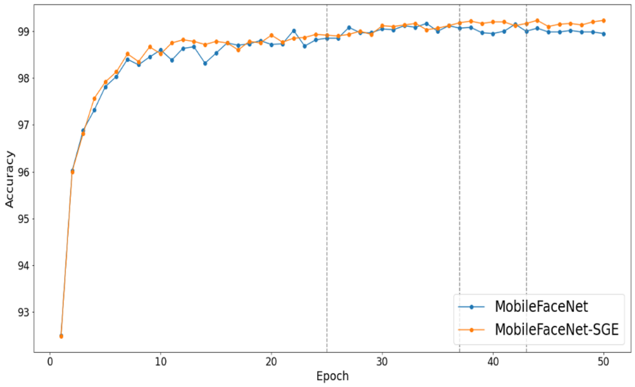


Fig. 16: Accuracy of face verification.

3) Test of face verification: We use the original MobileFaceNet and the MobileFaceNet-SGE to extract features, and use ArcFace as the loss function to train the network. Figure 16 shows the accuracies acheived by the two networks. It can be seen that for the first 37 epochs, their accuracies are quite similar, however, with the increase of the iteration, the accuracy of MobileFaceNet-SGE is higher than the original network. The accuracy is about 0.29% higher than the original network, from 98.95% to 99.23%, meanwhile the model size increased from 4.0MB increased to 4.2MB. It is acceptable to slightly increase the amount of parameters while improving performance.

4) System performance: In previous sections, we introduced the accuracy results of algorithms used in each module. As our proposed system is a combination of them, the high accuracy performance of each algorithm indicates that our system achieves a high security level. In this section, we will analysis the performance of two modes of the system.

For the cooperative authentication mode, the first level only use PIN code classification algorithm, so the accuracy is the same as the result of CNN-XGBoost hybrid model which is 96.42%. However, we allow the customer to repeat for 3 times. Each time can be regarded as an independent event. There are two possible results: the PIN code is correct or incorrect. The probability of all three times are incorrect can be calculated according to the Bernoulli formula [44]:

$$P_n\left(k\right) = C_n^k p^k \left(1-p\right)^{n-k} \left(k = 0, 1, 2, \cdots, n\right) \tag{3}$$

where $C_n^k = \frac{n!}{k!(n-k)!}$ is the binomial coefficient, n represents the repeat times, k stands for the times where the probability of incorrect p. So the final accuracy of the first level can be calculated using Equation 4. Based on Equation 3 and the accuracy of CNN-XGBoost hybrid model, the probability of incorrect $p = 1 - 96.42\% = 3.58\%$, both n and k equal to 3. As a result, the accuracy of the first level can be calculated as follows, which is very close to 100%.

$$\begin{aligned} L1_{acc} &= 1 - P_n\left(k\right) \qquad (4) \\ &= 1 - \frac{3!}{3!(3-3)!} 0.0358^3 \left(1 - 0.0358\right)^{3-3} \\ &= 1 - 0.0358^3 = 99.9954\% \end{aligned}$$

TABLE XII: Accuracy of second level.

	Voiceprint verification Correct (VoC)	Voiceprint verification Incorrect (VoI)
Face verification Correct (FaC)	96.98% × 99.23% = 96.23%	(1-96.98%) ×99.23% = 3.00%
Face verification Incorrect (FaI)	96.98% × (1-99.23%) = 0.75%	(1-96.98%) × (1-99.23%) = 0.02%

Meanwhile, the second level check both the cosine similarity of voiceprint and face features, the total accuracy of this level can be calculated using euqation 5, where VoC means Voiceprint verification Correct, VoI means Voiceprint verification Incorrect, FaC is short for Face verification Correct and FaI is abbreviation of Face verification Incorrect. The values are shown in Table XII. As a result, the accuracy of second level can be calculated as follows, which is as high as 99.98%.

$$\begin{aligned} L2_{acc} &= VoC * FaC + VoC * FaI + VoI * FaC \qquad (5) \\ &= 1 - FaI * VoI \qquad (6) \\ &= 1 - 0.02\% = 99.98\% \end{aligned}$$

Since the two levels are performed in serial order, we can calculate the accuracy of the Multi-Level Cooperative authentication mode (MLC_{acc}) using Equation 7, where the $L1_{acc}$ accounts for the accuracy of the first level and $L2_{acc}$ means the accuracy of the second level, and the final accuracy can be calculated as follows:

$$\begin{aligned} MLC_{acc} &= L1_{acc} * L2_{acc} \qquad (7) \\ &= 99.9954\% * 99.98\% = 99.9754\% \end{aligned}$$

From the result we can find out that, even though the accuracy of CNN-XGBoost hybrid model is low, the accuracy of first level is very high because 3 retries are allowed. To improve the delivery rate, anyone of the voiceprint and face identification pass, we think that the customer is correct. However, this is not safe enough, because it is easy to attack one of them instead of both. To improve the system security, we can change the strategy when both of them pass, the customer is identified as the correct one. In this situation, we should use Equation 8 to calculate the accuracy of the second level, and the final accuracy is $L1_{acc} * L2_{acc} = 99.9954\% * 96.23\% = 96.2256\%$.

$$\begin{aligned} L2_{acc} &= VoC * FaC \qquad (8) \\ &= 96.98\% * 99.23\% = 96.23\% \end{aligned}$$

For the non-cooperative authentication mode, we use face verification singly, so the accuracy of this mode is the same as face verification algorithm accuracy which is 99.23%.

VII. Conclusion

In this paper, we proposed and implemented a robotic delivery authentication system based on AI. A multi-level cooperative authentication scheme was proposed. The proposed method, while considering user convenience, the security is improved through multi-level authentication. For the first level, we allow the customers to repeat to achieve a high accuracy. However, we also realize that due to the influence of dialects, accents, etc., the reason for each failure may be the same, which cannot be solved by repetition. Therefore, the training dataset needs to be expanded so that a better model can be trained. Moreover, the second level can effectively protect the system in the case of the data leakage. For this level, there is a trade-off between security and successful delivery rate, different strategy should be used based on it. In order to deal with the case that the robot cannot classify the PIN code correctly due to the reasons such as a customer who has a strong accent, a non-cooperative scheme was proposed to detect and recognize the client. In order to speed up the calculation, we adopted lightweight models as the backbone. Our experimental results prove that the proposed algorithms have high classification and verification accuracy. Moreover, our non-cooperative recognition algorithm is more accurate than the original one. In conclusion, our proposed system is secure, accurate and efficient.

References

[1] J. Rheude, “Ecommerce growth from 2010 to 2020,” Red Stag Fulfillment, 09 2021. [Online]. Available: https://redstagfulfillment.com/2010s-ecommerce-growth-decade/

[2] M.-J. Lazar, “12 ecommerce shipping statistics to know in 2021 — readycloud,” Ready Cloud Suites, 04 2021. [Online]. Available: https://www.readycloud.com/info/your-guide-to-ecommerce-shipping-in-2021-2

[3] Metapack, “Ecommerce delivery benchmark report 2021,” Metapack, https://info.metapack.com/rs/700-ZMT-762/images/Ecommerce Delivery Benchmark Report 2021.pdf, Tech. Rep., 2021.

[4] D. Du, “Research on the application of “last-mile” autonomous delivery vehicles in the context of epidemic prevention and control,” in *2021 International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*. IEEE, 2021, pp. 74–77.

[5] “Online food delivery services global market report 2021: Covid-19 growth and change to 2030,” www.reportlinker.com, 04 2021. [Online]. Available: https://www.reportlinker.com/p06064489/Online-Food-Delivery-Services-Global-Market-Report-COVID-19-Growth-And-Change-To.html

[6] O. , “Last mile delivery: Costs, explanation, & how to optimize,” OptimoRoute, 09 2020. [Online]. Available: https://optimoroute.com/last-mile-delivery/

[7] G. Nichols, “The last mile: Robots take to streets for local delivery,” ZDNet, 03 2021. [Online]. Available: https://www.zdnet.com/article/the-last-mile-robots-take-to-streets-for-local-delivery/

[8] O. Bestsennyy, G. Gilbert, A. Harris, and J. Rost, “Telehealth: a quarter-trillion-dollar post-covid-19 reality,” *McKinsey and Company*, vol. 29, 2020.

[9] J. Buchan, A. Charlesworth, B. Gershlick, and I. Seccombe, “A critical moment: Nhs staffing trends, retention and attrition,” *London: Health Foundation*, 2019.

[10] S. Scott, “Meet scout: Field testing a new delivery system with amazon scout.” About Amazon, 01 2019. [Online]. Available: https://www.aboutamazon.com/news/transportation/meet-scout

[11] A. Heinla, “Starship completes one million autonomous deliveries,” Medium, 01 2021. [Online]. Available: https://medium.com/starshiptechnologies/one-million-autonomous-deliveries-milestone-65fe56a41e4c

[12] T. Yang, “Jd’s robot delivers first order in wuhan in coronavirus aid support,” JD Corporate Blog, 02 2020. [Online]. Available: https://jdcorporateblog.com/jds-robot-delivers-first-order-in-wuhan-in-coronavirus-aid-support/

[13] S. Kavirayani, D. S. Uddandapu, A. Papasani, and T. V. Krishna, “Robot for delivery of medicines to patients using artificial intelligence in health care,” in *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*. IEEE, 2020, pp. 1–4.

[14] A. Joy, R. Varghese, A. Varghese, A. M. Sajeev, J. Raveendran, A. Thomas, and K. Saran, “Medrobo medicine delivering and patient parameter monitoring robot,” in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2021, pp. 1808–1812.

[15] P. Manikandan, G. Ramesh, G. Likith, D. Sreekanth, and G. D. Prasad, “Smart nursing robot for covid-19 patients,” in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2021, pp. 839–842.

[16] I. Bakach, A. M. Campbell, and J. F. Ehmke, “A two-tier urban delivery network with robot-based deliveries,” *Networks*, 2021.

[17] M. Ostermeier, A. Heimfarth, and A. Hübner, “Cost-optimal truck-and-robot routing for last-mile delivery,” *Networks*, 2021.

[18] P. Gope, O. Millwood, and N. Saxena, “A provably secure authentication scheme for rfid-enabled uav applications,” *Computer Communications*, vol. 166, pp. 19–25, 2021.

[19] P. Gope and B. Sikdar, “An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 69, pp. 13 621–13 630, 2020.

[20] J. Yang, P. Gope, Y. Cheng, and L. Sun, “Design, analysis and implementation of a smart next generation secure shipping infrastructure using autonomous robot,” *Computer Networks*, vol. 187, p. 107779, 2021.

[21] K. Aravindhan and R. Karthiga, “One time password: A survey,” *International Journal of Emerging Trends in Engineering and Development*, vol. 1, no. 3, pp. 613–623, 2013.

[22] N. M. Haller, “The s/key (tm) one-time password system,” in *Symposium on Network and Distributed System Security*, 1994, pp. 151–157.

[23] G. Elert, “The nature of sound – the physics hypertextbook,” The Physics Hypertextbook, 2019. [Online]. Available: https://physics.info/sound/

[24] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[25] x. d. , “Xgboost documentation — xgboost 1.5.0-dev documentation,” xgboost.readthedocs.io. [Online]. Available: https://xgboost.readthedocs.io/en/latest/index.html

[26] Wikipedia contributors, “Boosting (machine learning) — Wikipedia, the free encyclopedia,” 2021, [Online; accessed 10-August-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Boosting_(machine_learning)&oldid=1033187774

[27] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “Ghostnet: More features from cheap operations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589.

[28] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[29] S. Chen, Y. Liu, X. Gao, and Z. Han, “Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices,” in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 428–438.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[31] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[32] X. Li, X. Hu, and J. Yang, “Spatial group-wise enhance: Improving semantic feature learning in convolutional networks,” *arXiv preprint arXiv:1905.09646*, 2019.

[33] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[34] (2016). [Online]. Available: https://docs.ray.io/en/ray-0.4.0/index.html

[35] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[36] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” *arXiv preprint arXiv:1411.7923*, 2014.

[37] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database forstudying face recognition in unconstrained environments,” in *Workshop on faces in’Real-Life’Images: detection, alignment, and recognition*, 2008.

[38] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[39] P. Warden, “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition,” *ArXiv e-prints*, Apr. 2018. [Online]. Available: https://arxiv.org/abs/1804.03209

[40] J. S. Garofolo, “Timit acoustic phonetic continuous speech corpus,” *Linguistic Data Consortium, 1993*, 1993.

[41] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

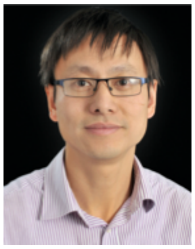
[42] H. Y. Khdier, W. M. Jasim, and S. A. Aliesawi, “Deep learning algorithms based voiceprint recognition system in noisy environment,” in *Journal of Physics: Conference Series*, vol. 1804, no. 1. IOP Publishing, 2021, p. 012042.

[43] J. Chang and D. Wang, “Robust speaker recognition based on dnn/i-vectors and speech separation,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5415–5419.

[44] J. W. Tukey *et al.*, *Exploratory data analysis*. Reading, Mass., 1977, vol. 2.

Wei Wang is currently working towards his MSc Degree in Cyber Security and Machine Learning at the University of Sheffield. He is interested in and currently researching Lightweight Authentication.

Prosanta Gope (Senior Member, IEEE) is currently working as an Assistant Professor with the Department of Computer Science (Cyber Security), University of Sheffield, U.K. He served as a Research Fellow with the Department of Computer Science, National University of Singapore. He has authored more than 75 peer-reviewed articles in several reputable international journals and conferences and has four filed patents. Several of his papers have been published in high impact journals, such as IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Industrial Electronics, and IEEE Transactions on Smart Grid. Primarily driven by tackling challenging real-world security problems, he has expertise in lightweight anonymous authentication, authenticated encryption, access control, security of mobile communications, healthcare, Internet of Things, Cloud, RFIDs, WSNs, smart-grid, and hardware security of the IoT devices. He received the Distinguished Ph.D. Scholar Award 2014 by National Cheng Kung University, Taiwan. He has served as the TPC Member/Chair in several reputable international conferences, such as IEEE TrustCom, IEEE GLOBECOM (Security-track), and ARES. He currently serves as an Associate Editor for the IEEE Internet of Things Journal, IEEE Systems Journal, IEEE Sensors Journal, Journal of Information Security and Applications (Elsevier), and the Security and Communication Networks.



Yongqiang Cheng is currently a Reader with the Department of Computer Science and Technology at the University of Hull, UK. His research interests include digital healthcare technologies, AI/ML, UAV, control theory and applications, embedded system and secure communication.